Reducing Complex Visualizations for Analysis

Lucas McDaniel
Institute of Northern Engineering
University of Alaska Fairbanks
lamcdaniel@gmail.com

Kara Nance
Department of Computer Science
University of Alaska Fairbanks
klnance@alaska.edu

Abstract

Data visualization provides a means to present known information in a format that is easily consumable and does not generally require specialized training. It is also well-suited to aid an analyst in discovering previously unknown information [1]. This is possible because visualization techniques can be used to highlight internal relationships and structures within the data, and present them in a graphical manner. Using visualization during the preliminary analysis phase can provide a pathway to enable an analyst to discover patterns or anomalies within the data that might otherwise go undiscovered as humans have an innate ability to visually identify patterns and anomalies.

Even when an analyst has identified a pattern or anomaly within the data, creating an algorithm that allows for automated detection of other occurrences of the same, or similar, patterns is a non-trivial task. While humans are innately skilled at pattern recognition, computers are not, and patterns that might be obvious for a human to identify might be difficult for a computer to detect even when assisted by a skilled analyst [2]. This paper describes a method of taking a complex visualization, and reducing it into several smaller components in order to facilitate computer analysis of the analyst-identified patterns or anomalies in the data. From there, a detection scheme can be generated through an analyst-supervised data analysis process in order to find more occurrences in a larger dataset.

Keywords: Data visualization, data analysis, visualization techniques

1. Introduction

Data visualization converts data into meaningful visual representations for human consumption through various techniques. These techniques can produce either static or dynamic (interactive) outputs, and the underlying mechanics for generating a graphical representation can differ significantly between techniques [3]. It is commonly used in cases where the volume of data being analyzed is more than can be reasonably done manually or without technological assistance. While many visualization techniques attempt to be data-agnostic – meaning the technique does not require ingest of specialized data – some domain-specific techniques do exist. For the purposes of this paper, we will focus on visualization techniques that are not domain-specific, and can be applied to arbitrary datasets in a general manner.

For any general text/numerical dataset, there are multiple ways it can be visualized with each visualization technique potentially capable of featuring unique internal structures or relationships within the data. Frequently, these techniques are only used to present some observation found through prior analysis. However, data visualization is not limited to this capacity and can be useful during the preliminary analysis phase by providing the analyst with other techniques to explore the data, and can often be done at lower costs than a traditional data-mining effort might require [4, 5]. By representing the data in a visual manner, we can enable the analyst to focus on tasks that humans are ideally suited for – anomaly and context identification, and pattern recognition – while allowing the computer to focus on tasks that it is suited for – large data processing.

Automated analysis of a dataset is a non-trivial process that can depend greatly on context, because what might appear as an anomaly in one case might be standard in another. For instance, the connection duration for several DNS queries is expected to be short, so a query that takes a long time might be an anomaly. However, a TCP connection to a webserver is expected to be long (relatively speaking), so a short connection could be an anomaly in this case. It is often easier for a human to utilize context and identify norms during the analysis process and use that basis to

URI: http://hdl.handle.net/10125/41894
ISBN: 978-0-9981331-0-2

identify expected norms, than it is to program a computer to perform the same analysis.

It is important to note that automated analysis of a dataset can be challenging, even after an analyst has identified an anomaly using a visualization technique. This is not unexpected. Once an analyst has identified an anomaly or pattern within a given visualization, it is a non-trivial process to determine what parts of the dataset produced this anomaly. More to the point, many visualization techniques are inherently lossy as the resolution of data is often reduced or similar data points are grouped together to reduce clutter in the resulting visualization. This increases the complexity of reversing the process. Even when the reversal can be completed, applying context to determine why this subset of the data is anomalous can still be difficult. The goal of generating a rule to find more occurrences can remain a challenging task to automate.

This paper will present a data-agnostic visualization technique to serve as a case study for several common limitations that can restrict a given visualization technique, and will identify methods that can be used to address these limitations. The ultimate goal of this effort is demonstrate how the process of using an analyst to identify patterns and anomalies in a complex visualization as a foundational step, and then breaking the visualization into smaller components can be beneficial. These smaller components can be analyzed with statistical or heuristic-based techniques to identify other components in the datasets (or other similar datasets). Analysis can then be conducted to determine the characteristics in this visualization technique that are important for this type of methodology, so that it can be applied to other types of visualization techniques.

2. Parallel Coordinates and Radar Charts

Parallel coordinates is a common technique for visualizing multivariate data where each dimension is orderable and bounded [6]. This technique works by drawing an even-width parallel axis for each dimension where the order of the axis can be user-defined or preset. Each record in a dataset has its coordinates mapped to the appropriate axis and a poly-line is drawn to connect them. See figure 1 for an example of parallel coordinates visualization.

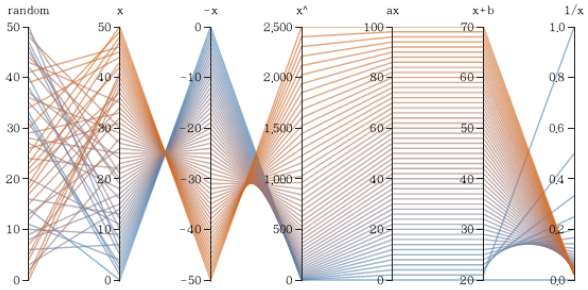

Figure 1: Parallel Coordinates Visualization [7]

Parallel coordinates are ideally suited for detecting anomalous records in large amounts of data as trends can be easily identified visually. For instance, the line segments between the columns *ax* and *x+b* in figure 1 are all horizontal and no lines cross. We as humans can easily see this, and if a new record were to be added that was not horizontal and crossed another line, it would be easy to identify as an anomaly. Similarly, in the last axis of figure 1, *1/x*, most records have a poly-line that terminates near 0 while only a few records do not. These potentially identify anomalies in the data that might merit further investigation. While it can be inferred from the axis headers what data is being visualized, it is important to realize that patterns and anomalies can be discovered without requiring this context, and these observations can serve as a starting point for further investigation.

Radar charts have some similarities with parallel coordinates, including the same restrictions on inputs. A primary difference is that they are designed to handle fewer records at a time. Radar charts tend to draw the parallel coordinate axis as spokes of a wheel with the lower bound of each axis generating a circle with radius 0 (or near 0), and the upper bound encircling the entire diagram. While it is not required to have each axis share the same scale, it is can be useful to have this type of sharing. See figure 2 for an example of a radar chart visualization.

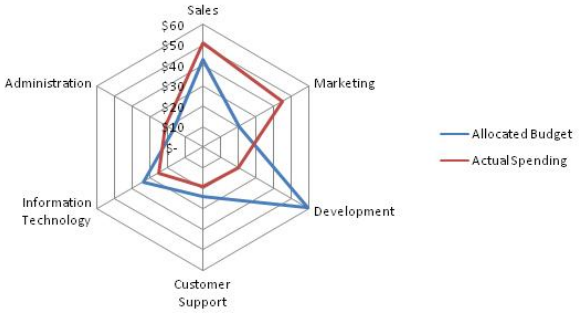

Figure 2: Radar Chart Visualization [8]

Radar charts are ideally suited for comparing a smaller number of records side-by-side. When a scale is chosen that places equal (or reasonable) weights on each axis, this type of visualization allows for quick comparison by approximating the area of the contained object as well as comparing relative differences on each axis. In figure 2, each axis indicates the cost of an activity with identical scaling (i.e., each step increases by $10). Based on this, it is easy to identify several categories where the allocated budget and the actual spending for the project differed significantly. Similarly, it is easy to estimate the areas encompassed by the poly-lines to determine that the total budget and spending for the project did not differ significantly.

3. Constraints and Customizations

It is worth noting that in addition to the explicit requirements for a given visualization technique there may also be implicit requirements to ensure the data works well with the visualization. Parallel coordinates works best when each dimension is not only orderable, but also when neighbors represent similarities and the scale for each axis produces a reasonable density. For instance, two network connections for similar lengths of time might indicate some type of similarity, but two connections to similar source ports will not have the same deeper meaning. A standard numeric mapping of all available destination ports will also tend to show most traffic going a very small range of available ports making differentiation challenging. It is often the case that preliminary statistical analysis to determine characteristics such as the density and distribution of the data may be needed to ensure appropriate mappings and scaling are chosen for a given visualization.

Radar charts have the same implicit restrictions on the data as well as a few additional ones. Specifically, when there are similarities between the axes, such as each axis represents a cost, using similar scaling can be useful. In this model, comparisons can be made on an axis-by-axis basis, and also on the area encompassed by the poly-lines. When such similarities don't exist, it can be common to split a radar chart back into a parallel coordinates diagram with few records to prevent analysts from comparing the areas encompassed.

As with any visualization technique, there are common customizations that can be made to make each technique work better with the data. Parallel coordinates customizations include:

- **Interactivity** – Regions on each axis can be selected to filter (or exclude) data.
- **Grouping** – Similar records can be grouped together to reduce clutter. Grouped records can be indicated by increasing the width of lines, increasing the color intensity, or by showing a distribution around the line by varying the alpha channel. [6]

Radar chart customizations include:

- **Grouping** – Similar records can be grouped in a single diagram. Using a unique color for each record can be used to show sets of records. Showing a distribution around the line by varying the alpha channel can be used to indicate grouped records (or uncertainty). [9]
- **Dynamic** – By taking a sequence of radar charts and sequencing them together, changes over time (or over some other field) can also be shown.

While there are many more types of common customizations that can be applied, this selection is sufficient to show the central need: to ensure the visualization technique presents upon a reasonable amount of data at any given time. Too much data and the outputs are untenable and too little means there may not be enough data to reach any reasonable conclusions. This observation can hold true for both human and computer analysts.

4. Complex Visualization Reduction

While a human analyst can identify anomalies within a dataset visualized through parallel coordinates either by using pattern recognition on the static image or through experimenting with interactive capabilities, automating this capability to extend it to a computer can be challenging. Fortunately, this process can be simplified by reducing a large parallel coordinates visualization into a series of radar charts. This type of reduction is beneficial for two reasons:

1. It is often the case that visualization techniques can be interactive (i.e., a user can customize these visualization in real time). However, an interactive visualization technique may not scale well with an arbitrary amount of data. Thus, having a technique where an analyst's work can be observed against a small dataset and subsequently automated across a larger dataset can lead to new insight.
2. By tying the analyst's process back to computational techniques, we can create algorithms for detecting similar patterns in

future datasets without requiring active involvement of a human analyst.

These observations can be applied to create a new analysis workflow such as the one described below for parallel coordinates and radar graphs:

1. An analyst unfamiliar with the dataset being analyzed examines a parallel coordinates visualization of the dataset.
2. Through interaction with the graph (such as selecting bounds on the axis or the resulting reduced radar charts), the analyst identifies sections of the data that are similar and that merit additional evaluation.
3. These identified sections are then used in a participatory training operation using traditional analysis techniques such as those discussed in the next section.
4. The techniques used (with weights and configurations identified during the training) are then applied to another similar dataset.
5. The results are used in a feedback loop to tune process with a goal of automated analysis without input from a human-analyst.

5. Computational Analysis Techniques

There are many types of standard analysis techniques and the appropriateness of each can vary depending on the type of analysis being conducted and the associated objectives. Data mining efforts commonly use clustering techniques in an attempt to identify structures within the data [10, 11, 12]. While analyzing specific techniques is not the purpose of this paper, we will briefly discuss tools and techniques that can be used to analyze multivariate data.

The *scikit-learn* Python library [13] provides easy access to several commonly used classification and clustering techniques, and served as the basis for the computation for this effort. The library works by taking a training dataset to learn how to classify the data through statistical or machine learning techniques, then can analyze new data to predict classifications. Since *scikit-learn* provides a common interface for each technique, it is trivial to switch between them to identify techniques that work well for a particular dataset and analysis objective. For the purposes of this effort, we selected the Naïve-Bayes classifier, which classifies the data based on probabilistic likelihood and is generally well-suited for supervised learning [16].

Supervised learning uses an oracle to answer a few questions, and then attempts to determine the motivation behind the oracle responses in order to predict the answer to future questions without additional input from the oracle. For the purposes of this project, the oracle is the human analyst and the question being answered is "which records are of interest?" Once an analysis technique has been trained based on these answers, it can then continue answering – or classifying – future records as interesting or uninteresting without the need for additional analyst involvement.

6. Empirical Results

From 1998 to 2000, DARPA released several datasets that could be used to train and test Intrusion Detection Systems [14]. These datasets consist of periodic process listings that could be used in a host-based system, as well as summaries of network connections for network-based systems. A network of hosts was subjected to various types of attacks over the course of several weeks, while standard business operations were conducted alongside these attacks. The data collected is broken up into several days, some of which could be used as training data (i.e., each network connection that was part of an attack is flagged as such and specifies the type of attack), and others as testing data.

The 1998 data set consists of nine weeks of data taken on Monday through Friday each week for a total of 45 days. For our test, we selected a single day – June 22, 1998 – from the testing set for our starting analysis. This singular day has over 10,000 recorded network connections as shown in the parallel coordinates visualization generated by *D3.js* in figure 3 [15]. While the data collected in a single day can be visualized in interactive parallel coordinates visualization through a standard browser on modern hardware, attempting to do the same for the full 45 days of data becomes challenging. When the dataset size is increased significantly, the browser will quickly become unresponsive, thus reemphasizing the need automating analyst-supervised analysis.

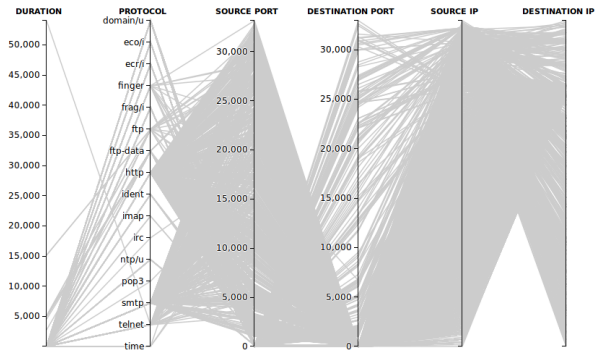

Figure 3: Parallel Coordinates – June 22, 1998

As discussed, parallel coordinates can be reduced to discrete radar charts. When these radar charts are flagged as interesting, they can be fed into an analysis technique to simplify automated analysis. For this analyst-supervised analysis, we selected and flagged a few radar charts, which demonstrated an easily identifiable anomalistic feature. Consider the example shown in figure 4. The radar charts show great similarity, but there are some anomalous characteristics that the human analyst is likely to identify. In figure 4, the analyst is likely to identify image C (and similar images) as anomalous.

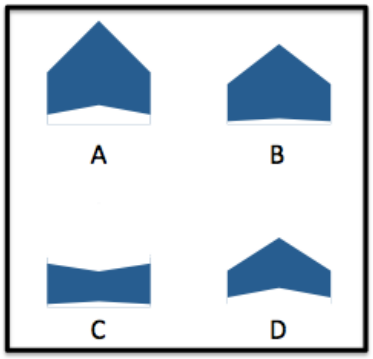

Figure 4: Radar chart anomalies

In our tests, about 20 radar charts with similar anomalies were selected during analyst-supervised identification phase (it is worth noting that the analyst is not required or expected to flag all occurrences within the testing data). Note that the analyst does not need to understand the data nor "why" the data is anomalous in all cases. The analyst with no knowledge of the datasets can identify anomalous images and then investigate the data to determine root cause. If the analyst has some information about the dataset, the process of understanding the "why" behind the anomalies will likely be less challenging.

A close-up an example radar chart from this phase with labeled axes is shown in figure 5. Close examination shows that the selected anomalous figure corresponds to connections that were uncharacteristically short for the given type of protocol. (Note that in the radar chart, string values are being mapped to integer values in order to utilize the same logarithmic scale as the numerical data.) While uncharacteristically short connections were chosen as the criteria, any other type of anomaly within the images could also have been selected.

It is also important to note that while the dataset flags connections involved in an attack as such, this information was not used during our analysis as it is would not be available in a realistic scenario since it would require a significant amount of human effort to classify the same number of records. The ultimate goal of this effort is to demonstrate how a small amount of human-lead analysis can be magnified through automation.

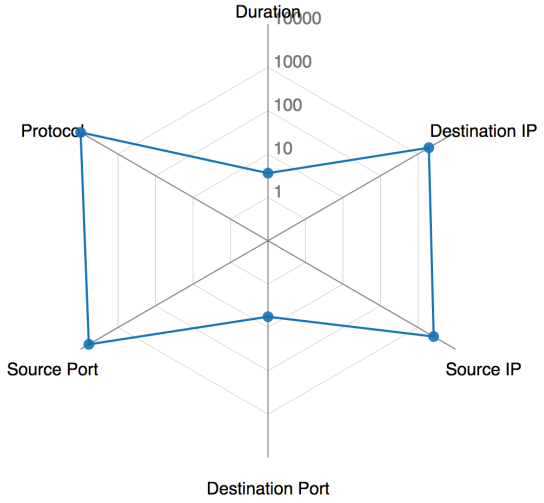

Figure 5: Radar Chart – June 22, 1998

The resulting data was then fed through a Naïve Bayes classifier for training, then tested against several days worth of data as shown in figures 6, 7 and 8. From these results, we can easily identify the criteria used during the analyst-supervised analysis phase: telnet sessions with a short duration. While there are many other techniques that could have been applied to find such occurrences, it should be noted that this process does not require any amount of technical skills from the analyst, as the classifier will attempt to

determine the criteria automatically. Figures 6, 7 and 8 clearly show that we were successfully able to determine the criteria used by the analysts during the supervision phase through use of the Naïve-Bayes classifier, and then automate this analysis across a larger set of data without need for additional analyst involvement.

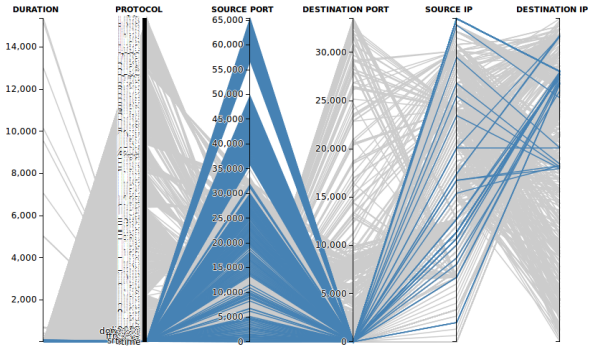

Figure 6: Parallel Coordinates – June 23, 1998

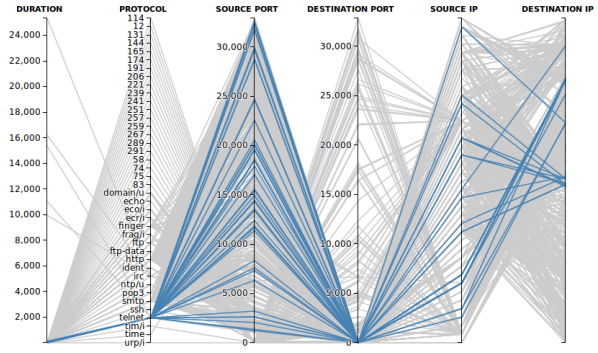

Figure 7: Parallel Coordinates – June 24, 1998

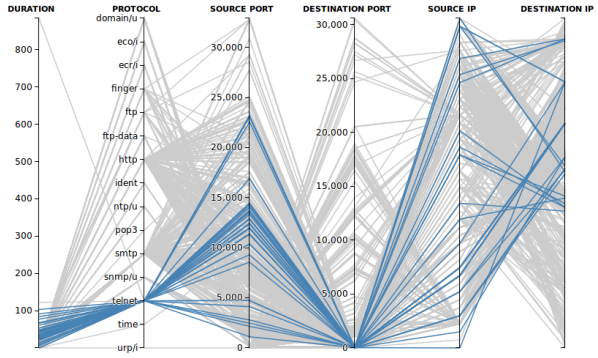

Figure 8: Parallel Coordinates – June 26, 1998

7. Conclusions

The overall objective of this effort was to determine if this method can improve the analysis process by reducing complex visualizations into a sequence of less computationally-intensive components. The analyst can identify anomalies. From there, a detection scheme can be generated through an analyst-supervised data analysis process in order to find similar occurrences in a larger dataset.

When discussing the efficacy of an analysis technique, there are two primary questions that need to be addressed:

- Are there interesting relationships in the dataset? (For instance, parallel coordinates will not always make it easy to identify interesting relationships. This can be seen in figure 1 where the relationships are simply mathematical operations.)
- Is it possible for this technique to identify these interesting relationships? (Similarly, using parallel coordinates to analyze a heap structure may not be useful either as the nature of the relationships cannot be adequately described with this technique.)

As this was a proof-of-concept research effort, these limitations were addressed in this limited scope by placing certain restrictions on the dataset, and ensuring that interesting information did exist in the datasets through prior analysis an screening. The choice of computational algorithm – Naïve Bayes classifier – used during this effort was made as the technique is commonly used in research as well as in practical applications.

This effort was able to successfully show how complex parallel coordinates visualization could be reduced to radar charts, thus allowing analyst-supervised learning techniques to be applied to automate analysis on future datasets. This required no specialized knowledge from the analyst to generate a classification technique, and only required the selection of 20 records of interest. While this workflow focused on parallel coordinates and used radar charts as the target reduction, this process could be extended to other visualization techniques that allow for such a reduction to be made.

8. References

[1] Fekete, Jean-Daniel, et al. "The value of information visualization." Information visualization. Springer Berlin Heidelberg, 2008. 1-18.

[2] Yang, Qiang, and Xindong Wu. "10 challenging problems in data mining research." International Journal of Information Technology & Decision Making 5.04 (2006): 597-604.

[3] Tory, Melanie, and Torsten Möller. "Rethinking visualization: A high-level taxonomy." Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on. IEEE, 2004.

[4] Keim, Daniel A. "Information visualization and visual data mining." Visualization and Computer Graphics, IEEE Transactions on 8.1 (2002): 1-8.

[5] Chen, Min, et al. "Data, information, and knowledge in visualization." Computer Graphics and Applications, IEEE 29.1 (2009): 12-19.

[6] Fua, Ying-Huey, Matthew O. Ward, and Elke A. Rundensteiner. "Hierarchical parallel coordinates for exploration of large datasets." Proceedings of the conference on Visualization'99: celebrating ten years. IEEE Computer Society Press, 1999.

[7] Yug. "Parallel coordinates-sample.png". 2015. Wikimedia Commons. June 11 2016.

[8] Clement, David. "Spider Chart2.jpg". 2006. Wikimedia Commons. June 11 2016.

[9] Cooke, R., and J. Van Noortwijk. "Graphical methods for uncertainty and sensitivity analysis." Sensitivity Analysis (2000): 245-266.

[10] Berkhin, Pavel. "A survey of clustering data mining techniques." Grouping multidimensional data. Springer Berlin Heidelberg, 2006. 25-71.

[11] Youngblood, G. Michael, and Diane J. Cook. "Data mining for hierarchical model creation." IEEE Transactions on Systems, Man, and Cybernetics, Part C 37.4 (2007): 561-572.

[12] Wu, Xindong, et al. "Top 10 algorithms in data mining." Knowledge and information systems 14.1 (2008): 1-37.

[13] "Scikit-learn." : Machine Learning in Python — 0.17.1 Documentation. N.p., n.d. Web. 15 May 2016.

[14] "DARPA Intrusion Detection Evaluation." MIT Lincoln Laboratory:. N.p., n.d. Web. 15 May 2016.

[15] @mbostock. "D3.js - Data-Driven Documents." D3.js - Data-Driven Documents. N.p., n.d. Web. 15 May 2016.

[16] Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." (2007): 3-24.